REMARKS

The foregoing amendments have been made to impart precision and clarity to the claims rather than to distinguish from prior art.

The abstract of the disclosure has been amended, as directed by the Examiner and in accordance with 37 CFR §1.72(b).

Applicants respectfully request reconsideration of this application as amended. Claims 1-36 are pending. Claims 1-22 and 24-36 are rejected. Claim 23 is objected to. Claim 36 is amended. Claims 37-46 are added.

The remaining comments are directed to the claims and in particular to Claim 1-22 and 24-36. The Office Action mailed on September 19, 2002 rejects Claims 1-22 and 28-34 under 35 U.S.C. 102(b) as alledgedly being anticipated by Frank et al., U.S. Patent 5,893,166.

The Examiner states with respect to Claims 1-12 and 31-34 that, "Frank discloses a method, executable code for transparently sharing virtual address translations, by accessing a translation and identifying if it is sharable...the protection status code identifies if the translation is sharable, and this procedure is transparent to the OS to the extent claimed, since once the shared space is designated the OS is not involved in the accessing/identification of a translation."

Applicant respectfully disagrees with the Examiners assertions. Claims 1 and 9, (as well as Claims 13, 20 and 35) are claiming operating-system transparent sharing of virtual address translations. Yet Frank discloses (e.g. Fig. 7, col. 9 lines 7-47 [emphasis added]) that:

> At step 410 of FIG. 7, upon receiving the create SSB command, the present invention causes a first process to perform a context switch into kernel mode. The mechanisms to switch into kernel mode within an VMS operating system (or privileged mode in Windows) are well known and any of a number of well known techniques can be used to perform step

410. Kernel mode of operating system 220 is a mode of operation that grants the first process special privileges to access and modify PTE data structures that are created and maintained by the operating system 220 in section 250a (FIG. 4A) of the system virtual address space.

At step 420, the first process accesses and reads certain data structures within the operating system 220 that indicate those pages of the system virtual address space that are not used by the operating system 220. Typically the operating system 220 consumes only a portion of the system virtual address space that is allocated for its use. This used portion, portion 250a of FIG. 4A, actively stores: 1) the program code (instructions or the kernel code) that make up the operating system 220; 2) various data structures (e.g., PTEs) provided for and used by each process which define the processes' virtual address space; 3) certain drivers and interfaces used internally by the operating system 220; and 4) a general and small extra space that is reserved for operating system use only.

In one exemplary operating system, used portion 250a consumes only 25 percent of the total 2.0 GB allocated for system virtual memory space. The remainder of the system virtual memory space, e.g., portion 250b, is unused system virtual memory space. When physical memory is allocated to portion 250a of the system virtual address space, then individual entries into PTE table 310 (FIG. 5) are added to map pages this used memory space. At step 420, one implementation of the present invention, accesses this PTE table 310 to then determine those portions of the system virtual address space that are not mapped by the operating system and therefore are unused.

At step 425 of FIG. 7, the present invention assigns (e.g., links together) these identified unused portions of the system virtual address space to create the SSB, portion 250b, of the present invention.

Applicants respectfully submit that Frank does not disclose operating-system transparent sharing of virtual address translations. Claims 1, 9, 13 and 20 further claim transparently identifying if the virtual address translation is sharable or the providing of a first sharing indication. If the protection status code identifies whether the translation is sharable as asserted by the Examiner, then since Frank discloses (e.g. col. 10, lines 24-33 [emphasis added]) that:

> At step 430 of FIG. 7, the present invention accesses entries 340a to 340f and in column 315, sets the protection status code to indicate the lowest privilege mode so that processes 1 to n can access the SSB portion 250b.

> PTE table 310 is a data structure stored in system virtual space 250a, so at step 430, the present invention updates the contents of memory portion 250a to perform the above PTE status modification.
>
> At step 440 of FIG. 7, the present invention then performs a context switch out of kernel mode.

the protection status code is set to indicate sharability while in kernel mode, rather than transparent to the operating system. Thus, Claims 1, 9, 13 and 20 are not anticipated by Frank.

On the other hand, Claim 31 sets forth a processor comprising: an address translation stage including a storage array having... entries to associate virtual address data with corresponding translated address data; a first storage element to store a first translated address data...; and a control logic to identify a sharability status for the first translated address data and to provide a first sharing indication to indicate if the first entry may be shared.

Frank does not disclose a processor, but rather "uses a process to switch to kernel mode, then identifies those sections of the operating system virtual memory space that are not being used;... alters the privileges of the PTEs... so that user mode processes can access... operating system virtual memory... The result is a statically mapped large memory... buffer," (abstract lines 10-18).

Therefore, Applicant respectfully submits that Claims 1, 9, 13, 20 and 31 are patently distinguished over the art cited by the Examiner. Applicant further believes that Claims 2-8, 10-12, 14-19, 21-22, 28-30 and 32-34 being dependent therefrom are also patentable. Applicant respectfully requests the Examiner withdraw his rejection under 35 U.S.C. 102(b).

The Office Action rejects Claims 24-27 and 35-36 under 35 U.S.C. 103(a) as allegedly being unpatentable over Frank et al., in view of Schimmel, U.S. Patent 6,105,113.

The Examiner argues that the consistency techniques of Schimmel include the incorporation of a shared indication when applied to the Frank system, because typical cache consistency techniques include shared indications and that it would have been obvious to add a TLB storing the shared indications to improve the translation speed and provide translation consistency.

Applicant respectfully disagrees. Schimmel does not disclose a shared indication for TLBs or storing a shared indication in a TLB. Further, if as the Examiner has suggested, the protection status code of Frank identifies whether the translation is sharable, then would the combination of Frank and Schimmel result in the sharing of all "user mode" TLBs between processes? How does the Examiner propose that such a combination would provide, for example, dynamic paging and non-conflicting virtual addresses for ordinary processes in user mode?

On the contrary, what Schimmel has disclosed (e.g. col. 4, lines 22-43 [emphasis added]) is:

> Regardless of which protocol is employed, the TLB snooping controller snoops the cache-memory interconnect and detects the update or invalidate signal and the PTE address. The TLB search engine searches the PTE address tag of the TLB table for the PTE address. If the PTE address is found in the TLB table, the associated translation is updated or invalidated by the TLB updating module. Thus, when the operating system changes a page table entry, translations in TLBs are automatically kept consistent.
>
> The present invention can be implemented on any computer system that employs virtual memory. Thus, the present invention can be implemented in both uni-processor environments and multiple processor environments. The present invention is especially useful in shared memory, multi-processor systems where page migration occurs. Shared memory systems that benefit from the present invention include centralized shared memory systems, such as, symmetric multiple processor (SMP) systems and distributed shared memory (DSM) systems. The present invention can be employed to maintain consistency for any number of TLBs in a system.

Therefore, since Frank discloses (e.g. col. 9, lines 55-65 [emphasis added]) that:

> Within step 425, the present invention links together the pages of the SSB by constructing page table entries within PTE table 310. This is accomplished by assigning free pages of the physical memory map 270 to the unused pages of the system virtual memory as identified in step 420. By performing this action, the present invention creates a memory map of portion 250b to range 258 of the physical memory map 270 (FIG. 4A). Within the present invention, this map is static in that it is statically mapped to physical pages of memory and not thereafter updated or altered while the corresponding SSB is valid.

and since the SSB and associated PTE is set up once prior to use and not thereafter updated or altered, it would seem that the only time Schimmel's disclosed consistency mechanism functions in connection with Franks disclosed method is after the SSB and associated PTE have been invalided.

Accordingly, in light of the above arguments, Applicant respectfully submits that combining Frank and Schimmel to add a TLB storing the shared indications would not have been obvious. Therefore Claims 24-27 and 35-36 are also patently distinguished over the art cited by the Examiner. Applicant respectfully requests the Examiner withdraw his rejection under 35 U.S.C. 103(a).

Applicants respectfully submit that added Claims 37-46 particularly point out and distinctly claim novel subject matter, which Applicants regard as their invention. Applicant further believes that Claims 37-39 and 40-46 being dependent from Claims 36 and 35 respectively are also patentable.

Applicants, therefore, believe that Claims 1-36 and Claims 37-46 are presently in condition for allowance and such action is earnestly solicited.

CONCLUSION

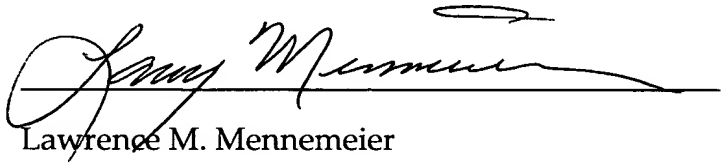
Applicants respectfully submit the present claims for allowance. If the Examiner believes a telephone conference would expedite or assist in the allowance of the present application, the Examiner is invited to call Lawrence M. Mennemeier at (408) 765-2194.

Authorization is hereby given to charge our Deposit Account No. 02-2666 for any charges that may be due.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN

Date: 1-20-03



Lawrence M. Mennemeier

Reg. No. 51,003

12400 Wilshire Boulevard

Seventh Floor

Los Angeles, CA 90025-1026

(408) 720-8300

VERSION WITH MARKINGS TO SHOW CHANGES MADE

IN THE ABSTRACT:

The abstract on page 36 of the present application, is replaced by the amended abstract on the separate sheet attached.

IN THE SPECIFICATION

The heading on page 28 is replaced as follows:

[IN]THE CLAIMS

IN THE CLAIMS

Claim 36 is amended as follows:

36. (Amended once) A multithreading processor comprising:

an address translation stage including a translation lookaside buffer having a plurality of entries to translate virtual addresses to physical addresses;

a first entry of the plurality of entries to translate a first virtual address for a first process;

a control logic comprising circuitry to identify a sharability of the first entry [process];

the control logic further to provide a first sharing indication to indicate if the first entry may be shared by a second process; and

a sharing indication field in the first entry to store the first sharing indication provided by the control logic.

Claims 37-46 are added.

37-46 (New)